

CLAIMS

What is claimed is:

- 1 1. A method comprising:
2 retrieving, by a request unit, a number of requests for security operations from a
3 host memory, wherein the number of requests are in an order within the host memory;
4 distributing, by the request unit, the number of requests for the security operations
5 to a number of execution units, wherein the distribution is based on availability of the
6 number of execution units;
7 processing the number of requests for the security operations by the number of
8 execution units; and
9 outputting results of the number of requests for the security operations to locations
10 within the host memory, wherein an order of outputting of the results can be different
11 from the order of the requests within the host memory.
- 1 2. The method of claim 1, wherein the number of requests relate to a number of
2 different secure sockets layer sessions, wherein the number of requests can be
3 independent of one another.
- 1 3. The method of claim 1, wherein a time to process one of the number of requests
2 can be different from a time to process another of the number of requests.
- 1 4. The method of claim 1, wherein each of the number of execution units can process
2 any of the different types of the security operations within the number of requests.
- 1 5. The method of claim 1, wherein a request can include a macro security operation.

1 6. The method of claim 1, wherein retrieving of the number of requests comprises
2 retrieving the number of requests from a queue in the host memory using a first direct
3 memory access (DMA) read operation for the number of requests and retrieving input
4 data structures related to the number of requests from the host memory using a second
5 DMA read operation.

1 7. The method of claim 6, wherein outputting the results of the number of requests
2 comprises outputting the results of the number of requests using a DMA write operation.

1 8. The method of claim 7, wherein retrieving of one of the number of requests and
2 outputting of the one of the number of requests includes approximately two DMA
3 operations.

1 9. The method of claim 1, wherein the outputting of the results of the number of
2 requests for the security operations to locations within the host memory includes
3 outputting the results of the number of requests for the security operations to locations
4 within the host memory based on pointers stored in the number of requests.

1 10. The method of claim 1, wherein outputting results of the number of requests for
2 the security operations to locations within the host memory includes setting a value of a
3 completion code at the locations within the host memory, the completion code to indicate
4 completion of the request.

1 11. A method executing on a host processor, the method comprising:
2 storing a number of requests for security operations within a request queue within
3 a host memory, wherein the number of requests are in an order within the request queue;
4 storing data related to the number of requests for security operations into a
5 number of input data structures within the host memory;

6 allocating a number of output data structures within the host memory, wherein a
7 coprocessor is to write results of the number of requests for the security operations into
8 the number of output data structures, wherein the coprocessor can write the results in an
9 order that is different from the order of the requests within the request queue; and
10 for each of the number of requests, allocating a thread for execution on the host
11 processor, wherein the thread periodically checks a value of a completion code stored in
12 the output data structure for the associated request, the completion code to indicate that
13 the request is completed by the coprocessor.

1 12. The method of claim 11, wherein the request queue is locked during the storing of
2 the number of requests within the request queue.

1 13. The method of claim 11, further comprising writing a number of the number of
2 requests stored within the request queue into a memory location on the coprocessor.

1 14. The method of claim 11, wherein each thread of execution is associated with a
2 different security socket layer session.

1 15. The method of claim 11, wherein the thread of execution checks the value of the
2 completion code for the associated request independent of other threads of execution.

1 16. The method of claim 11, wherein at least one of the number of requests include a
2 macro security operation.

1 17. A processor comprising:
2 a number of execution units to process a number of requests for security
3 operations, wherein the number of execution units are to output the results of the number
4 of requests to a number of output data structures associated with the number of requests
5 within a remote memory based on pointers stored in the number of requests and wherein

6 the number of execution units can output the results in an order that is different from the
 7 order of the requests in a request queue; and
 8 a request unit coupled to the number of execution units, wherein the request unit is
 9 to retrieve a portion of the number of requests from the request queue within the remote
 10 memory and associated input data structures for the portion of the number of requests
 11 from the remote memory and wherein the request unit is to distribute the retrieved
 12 requests to the number of execution units based on availability for processing by the
 13 number of execution units.

1 18. The processor of claim 17, wherein the number of requests relate to a number of
 2 different secure sockets layer sessions, wherein the number of requests can be
 3 independent of one another.

1 19. The processor of claim 17, wherein a time to process one of the number of
 2 requests can be different from a time to process another of the number of requests.

1 20. The processor of claim 17, wherein each of the number of execution units can
 2 process any of the different types of the security operations within the number of
 3 requests.

1 21. The processor of claim 17, wherein a request can include a macro security
 2 operation.

1 22. The processor of claim 17, wherein the request unit is to retrieve the number of
 2 requests from the request queue using a first direct memory access (DMA) read operation
 3 and wherein the request unit is to retrieve the number of associated input data structures
 4 using a second DMA read operation.

1 23. The processor of claim 22, wherein the number of execution units are to output
2 the results of the number of requests to the number of output data structures using a DMA
3 write operation.

1 24. A system comprising:
2 a host processor coupled to a system bus;
3 a host memory coupled to the system bus, wherein the host memory is to store,
4 a request queue to include a number of requests for security operations,
5 wherein the number of requests are in an order within the request queue;
6 a number of input data structures to include data associated with the
7 number of requests; and
8 a number of output data structures to include results from processing of the
9 number of requests; and
10 a coprocessor coupled to the system bus, wherein the coprocessor comprises,
11 a number of execution units to process the number of requests for security
12 operations, wherein the number of execution units are to output the results of the number
13 of requests to the number of output data structures associated with the number of requests
14 based on pointers stored in the number of requests and wherein the number of execution
15 units can output the results in an order that is different from the order of the requests in
16 the request queue; and
17 a request unit coupled to the number of execution units, wherein the
18 request unit is to retrieve a portion of the number of requests from the request queue and
19 associated input data structures for the portion of the number of requests from host
20 memory and wherein the request unit is to distribute the retrieved requests to the number
21 of execution units based on availability for processing by the number of execution units.

1 25. The system of claim 24, wherein the number of requests relate to a number of
2 different secure sockets layer sessions, wherein the number of requests can be
3 independent of one another.

1 26. The system of claim 24, wherein a time to process one of the number of requests
2 can be different from a time to process another of the number of requests.

1 27. The system of claim 24, wherein each of the number of execution units can
2 process any of the different types of the security operations within the number of
3 requests.

1 28. The system of claim 24, wherein a request can include a macro security operation.

1 29. The system of claim 24, wherein the request unit is to retrieve the number of
2 requests from the request queue using a first direct memory access (DMA) read operation
3 and wherein the request unit is to retrieve the number of associated input data structures
4 using a second DMA read operation.

1 30. The system of claim 29, wherein the number of execution units are to output the
2 results of the number of requests to the number of output data structures using a DMA
3 write operation.

1 31. A machine-readable medium that provides instructions, which when executed by a
2 machine, causes the machine to perform operations comprising:
3 retrieving, by a request unit, a number of requests for security operations from a
4 host memory, wherein the number of requests are in an order within the host memory;

5 distributing, by the request unit, the number of requests for the security operations
6 to a number of execution units, wherein the distribution is based on availability of the
7 number of execution units;
8 processing the number of requests for the security operations by the number of
9 execution units; and
10 outputting results of the number of requests for the security operations to locations
11 within the host memory, wherein an order of outputting of the results can be different
12 from the order of the requests within the host memory.

1 32. The machine-readable medium of claim 31, wherein the number of requests relate
2 to a number of different secure sockets layer sessions, wherein the number of requests can
3 be independent of one another.

1 33. The machine-readable medium of claim 31, wherein a time to process one of the
2 number of requests can be different from a time to process another of the number of
3 requests.

1 34. The machine-readable medium of claim 31, wherein each of the number of
2 execution units can process any of the different types of the security operations within the
3 number of requests.

1 35. The machine-readable medium of claim 31, wherein a request can include a macro
2 security operation.

1 36. The machine-readable medium of claim 31, wherein retrieving of the number of
2 requests comprises retrieving the number of requests from a queue in the host memory
3 using a first direct memory access (DMA) read operation for the number of requests and
4 retrieving input data structures related to the number of requests from the host memory
5 using a second DMA read operation.

1 37. The machine-readable medium of claim 36, wherein outputting the results of the
2 number of requests comprises outputting the results of the number of requests using a
3 DMA write operation.

1 38. The machine-readable medium of claim 37, wherein retrieving of one of the
2 number of requests and outputting of the one of the number of requests includes
3 approximately two DMA operations.

1 39. The machine-readable medium of claim 31, wherein the outputting of the results
2 of the number of requests for the security operations to locations within the host memory
3 includes outputting the results of the number of requests for the security operations to
4 locations within the host memory based on pointers stored in the number of requests.

1 40. The machine-readable medium of claim 31, wherein outputting results of the
2 number of requests for the security operations to locations within the host memory
3 includes setting a value of a completion code at the locations within the host memory, the
4 completion code to indicate completion of the request.

1 41. A machine-readable medium that provides instructions, which when executed by a
2 host processor, causes the host processor to perform operations comprising:
3 storing a number of requests for security operations within a request queue within
4 a host memory, wherein the number of requests are in an order within the request queue;
5 storing data related to the number of requests for security operations into a
6 number of input data structures within the host memory;
7 allocating a number of output data structures within the host memory, wherein a
8 coprocessor is to write results of the number of requests for the security operations into
9 the number of output data structures, wherein the coprocessor can write the results in an
10 order that is different from the order of the requests within the request queue; and

11 for each of the number of requests, allocating a thread for execution on the host
12 processor, wherein the thread periodically checks a value of a completion code stored in
13 the output data structure for the associated request, the completion code to indicate that
14 the request is completed by the coprocessor.

1 42. The machine-readable medium of claim 41, wherein the request queue is locked
2 during the storing of the number of requests within the request queue.

1 43. The machine-readable medium of claim 41, further comprising writing a number
2 of the number of requests stored within the request queue into a memory location on the
3 coprocessor.

1 44. The machine-readable medium of claim 41, wherein each thread of execution is
2 associated with a different security socket layer session.

1 45. The machine-readable medium of claim 41, wherein the thread of execution
2 checks the value of the completion code for the associated request independent of other
3 threads of execution.

1 46. The machine-readable medium of claim 41, wherein at least one of the number of
2 requests include a macro security operation.